

ADAPTIVE INVESTMENT STRATEGIES FOR PERIODIC ENVIRONMENTS

J.-Emeterio Navarro

Arbeitsgruppe Künstliche Intelligenz
Institut für Informatik
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin, Germany

Abstract

In this paper, we present an adaptive investment strategy for environments with periodic returns on investment. In our approach, we consider an investment model where the agent decides at every time step the proportion of wealth to invest in a risky asset, keeping the rest of the budget in a risk-free asset. Every investment is evaluated in the market via a stylized return on investment function (RoI), which is modeled by a stochastic process with unknown periodicities and levels of noise. For comparison reasons, we present two reference strategies which represent the case of agents with zero-knowledge and complete-knowledge of the dynamics of the returns. We consider also an investment strategy based on technical analysis to forecast the next return by fitting a trend line to previous received returns. To account for the performance of the different strategies, we perform some computer experiments to calculate the average budget that can be obtained with them over a certain number of time steps. To assure for fair comparisons, we first tune the parameters of each strategy. Afterwards, we compare the performance of these strategies for RoIs with different periodicities and levels of noise.

1 Introduction

Finding a proper investment strategy is a problem that has been addressed by many researchers from different areas. In economy, this problem usually concerns the behavior that an investor should follow in order to maximize the profits under an uncertain environment. To this end, researchers usually investigate the relation between methods for optimization under uncertainty, the different preferences of an investor and the amount of information available from the environment. For this, different measures of risk aversion have been proposed together with the classification of investors by their behavior towards risk (e.g. risk-averse, risk-neutral or risk-seeking behaviors), see [3]. Many researchers have been also concerned in finding different manners to control the *risk-exposure*. Many of the proposed methods are based on decision-making and utility theory and are addressed to scenarios where the investor can choose between investing in a risky or a risk-free asset, see [4, 18, 19, 20]. And other researchers have extended this to the problem of portfolio diversification, where more than one risky asset is considered, see [26, 27, 28].

On the other hand, many researchers have used different machine learning methods to find good investment strategies in different type of stochastic environments. For example, in [25] the authors use neural networks

to find patterns from financial time series, where the main goal is to find changes in volatility. And in [11], the authors propose the use of a risk-sensitive reinforcement learning algorithm to find the most proper policy for controlling under constraints and applied it to the control of a feed tank with stochastic inflows. Other techniques from machine learning that are frequently used for investment decision problems are those based on *evolutionary computation*. For example, those using *genetic programming* and *genetic algorithms* for portfolio management, inducing rules for bankruptcy prediction, and assigning credit scoring, see [6]. Some investment strategies based on genetic programming techniques usually lead to profitable trading strategies, however, they usually find strategies which are difficult to understand and sometimes they cannot be funded [17, 33, 36, 37]. Even though investment strategies that are based on genetic algorithms may be also difficult to abstract and to explain, we believe that they are more natural and understandable than those using genetic programming techniques [7]. However, many of these approaches are applied to environments that are stationary; this means that some of them cannot be directly applied to changing environments. In the literature, there are some researches which have investigated the use of genetic algorithms in changing environments [5, 14]. However, to our knowledge, they have not been applied specifically to the problem of controlling the proportion of investment in periodic environments.

On the other hand, the typical scenario to study investment strategies is to let an agent choose between betting in a lottery or receiving a constant amount of money [2]. This simple scenario is usually extended to different type of investment models where investors are commonly referred to as agents and the complexity of the investment models may differ considerably, see [8, 22, 24]. In some of these models the amount of money that the agents invest in the market is assumed to be proportional to their budget, this assumption is also called investment fraction or investment proportion. Researchers investigate in these models from the optimal investment strategies to the different properties that emerge in the artificial market, see [27, 28]. Interestingly, if the market is simply treated as a random variable and the proportion of investment is fixed to a constant value, then it has been shown that eventually the agent loses all its money in the course of time [31, 38]. In order to avoid bankruptcy, the agent may have an income, [21, 34, 38], or a budget-barrier may be assumed [23]. Some researchers have investigated different strategies to control the proportion of investment in this type of models for different scenarios [28, 32]. On the other hand, some other authors use different artificial market models to compare the performance of agents with zero-intelligence and rational agents [9, 12].

This paper may also draw interest on the research area of pattern recognition of time series. In particular, for the cases when there is no prior knowledge of the existence of a periodic signal or of its characteristics, see [1, 39]. Note that with some small proper changes on the proposed adaptive algorithm, a useful algorithm could be proposed for the detection and measurement of periodic signal in time series.

In this paper, we propose a new approach based on evolution for finding investment strategies in periodic environments. This paper is organized as follows. Section 2 describes the investment model where the agent decides at every time step the percentage of wealth to invest in a risky asset keeping the rest in cash. Section 3 presents an adaptive investment strategy based on a *Genetic Algorithm* (GA) for environments with periodic time series. In Section 4 we present the results obtained for different computer experiments. We decided to perform our computer experiments in a controlled scenario where the dynamics of the environment are known. For that reason, we assume that the risky asset is modeled by a stochastic process with changing periodicity and different levels of noise, i.e. stylized exogenous returns, presented in Section 4.1. For the sake of completeness, we compare the performance of the adaptive strategy proposed in this paper with other investment strategies which are discussed in Section 4.2. For our comparison, we include two reference strategies which represent the agent with both zero-knowledge and complete-knowledge of the dynamics of the RoI. We also include an investment strategy based on technical analysis that basically forecasts the next return by fitting a trend line to the previous received returns. To account for the performance of the different strategies, we perform some computer experiments to calculate the average budget that can be obtained with them over a certain number of

time steps. The experiments to account for the performance of the investment strategies are divided into two sections. First, in Section 4.3 we consider a stationary environment, i.e. returns with fixed periodicity, and in Section 4.4 we consider a non-stationary environment, i.e. return with changing periodicity. To assure for fair comparisons, we first tune the parameters of each strategy and afterwards, we compare their performance for returns with different levels of noise.

2 Investment Model

We consider an investment model [31, 32] where an agent is characterized by two individual variables: (i) its *budget* $x(t)$, i.e. its wealth and (ii) its *investment proportion* $q(t)$, i.e. its attitude towards risk in a market. The budget, $x(t)$, changes in the course of time t by means of the following dynamic:

$$x(t+1) = x(t) \left[1 + r(t) q(t) \right] \quad (1)$$

More in detail, this means that the agent at time t invests a portion $q(t)x(t)$ of its total budget. And this investment yields a gain or loss on the market, expressed by $r(t)$, the return on investment, *RoI*.

Some authors assume that returns are obtained by means of continuous double auction mechanisms [22, 24], however, in our approach, we rather consider that the returns are not being influenced by agent's actions. In other words, we assume that the agent has a small budget and its actions do not affect the evolution of the returns. Later on, we present more in detail the dynamic for the returns with seasonal market changes, Eq. (4).

The behavior of the agent in this environment is expressed in terms of its investment proportion, $q(t)$, which corresponds to the percentage or portion of agent's budget that is susceptible to win or lose, i.e. the agent's attitude towards risk. We assume that the agent's investment proportion may change for example, dependent on the agent's predictions or assumptions about the market dynamics.

Since $q(t)$ always represents a portion of the total budget $x(t)$, it is bound to a minimum value of zero and a maximum value of one, i.e. $q(t) \in [0, 1]$. This means that an agent with $q(t) = 0$ decides at time step t to perform no investments at all, whereas an agent with $q(t) = 1$, is investing at time step t all its capital. For the sake of completeness, we assume that the minimal and maximal investment-proportions are described by q_{\min} and q_{\max} , respectively.

Thus, in this paper we present an adaptive investment strategy, expressed by a method to find the most proper $q(t)$ and we focus on the performance of this investment strategy in periodic returns. Of course the agent may have some bounded memory about past RoI that could be used for predictions of future RoI. And as mentioned above, we assume a simple dynamic for the returns allowing us to focus in the feedback of these market returns on the investment strategy (and not on the feedback of the strategies on the market).

Last but not least, we assume that the agent invests independently in the market, i.e. there is no direct interaction with other agents.

3 Adaptive Investment Strategy

In this section, we present an adaptive investment strategy based on a *Genetic Algorithm* for controlling proportions of investment in periodic environments. For simplicity, we call this strategy *Genetic Algorithm for Changing Environments* (GACE).

Genetic algorithms (GA) are stochastic search algorithms based on evolution that explore progressively from a large number of possible solutions finding after some generations the best solution for the problem. Inspired by natural selection, these powerful techniques are based on some defined evolution operators, like selection, crossover and mutation [10, 13, 16, 29].

In our approach, we consider that an agent uses a GA to find the most proper set of investment proportions for every time step. For this, we show on the following the specifications for the GA.

3.1 Encoding Scheme

A population of chromosomes $j = 1, \dots, C$, where each chromosome j has an array of genes, g_{jk} , where $k = 0, \dots, G_j - 1$, and G_j is the length of the chromosome j . The length of a chromosome is assumed to be in the range $G_j \in (1, G_{\max})$, where G_{\max} is a parameter that specifies the maximal allowed number of genes in a chromosome. The values of the genes could be binary, but for programming reasons we use real values, see [29]. Moreover, each chromosome j represents a *set of possible strategies* of an agent, i.e. each g_{jk} corresponds to an investment proportion.

3.2 Fitness Evaluation

Each chromosome j is evaluated after a given number of time steps by a *fitness function*, $f_j(\tau)$, which is defined as follows:

$$f_j(\tau) = \sum_{k=0}^{G_j-1} r(t) g_{jk} ; \quad k \equiv t \bmod G_j, \quad (2)$$

where τ is a further time scale in terms of generations. When a generation is completed, the chromosomes' population is replaced by a new population of better fitting chromosomes with the same population size C .

As you can see, every g_{jk} is multiplied by a different value of $r(t)$ in the course of time. Since the fitness of a chromosome tends to be maximized, negative $r(t)$ should lead to small values of g_{jk} , i.e. small investment proportions. On the other hand positive $r(t)$ should lead to larger values of g_{jk} , i.e. large investment proportions. Because of this, we consider the product of $r(t)g_{jk}$ as a performance measure, which is in accordance with our investment model, Eq. (1). Noteworthy, in this approach the GA tries to find the chromosomes leading to larger profits. Another different approach would be to implement a GA to find the chromosomes that minimize the loss, in which case, we would have a different fitness function. Also note that chartists usually study the past movement of stock prices; however, this approach differs from ours in the fact that we treat directly returns on investment and not price movements.

3.3 Selection of a New Population

If we assume that chromosomes have fixed length, $G_j = G_{\max}, \forall j$, then the most proper number of time steps, t_{eval} , that have to elapse in order to evaluate all chromosomes' genes is $t_{\text{eval}} = G_{\max}$. In other words, the number of time steps needed to evaluate the population is equal to the fixed length of the chromosomes.

Moreover, it can be shown that the population converges faster towards optimal investment proportions if the length of the chromosomes is equal to the periodicity of the returns, $G_{\max} = T$.

However, this previous assumption corresponds to the ideal case where the agent knows a priori the periodicity of the returns and sets the length of all chromosomes to the value of the periodicity, hence the agent selects a new population after all genes of all chromosomes are being evaluated. Thus, if the chromosomes have different length the question now is the following: *After how many time steps, t_{eval} , a new generation of chromosomes should be obtained?* In the following, we propose different approaches to answer this question.

3.3.1 Time steps for evaluation

Different approaches can be proposed to determine the number of time steps t_{eval} that should elapse to select a new generation of chromosomes. As mentioned above, the simplest approach, called *GMaximum*, is to select a new population after a fixed number of time steps. If G_{max} is equal to the maximal length of the chromosomes, $t_{\text{eval}} = G_{\text{max}}$, then all chromosomes' genes in the population will be evaluated. However, such an approach leads to slow convergence of the population. A different approach may be to choose the number of time steps for evaluation accordingly to the length of the best chromosome in the population. This approach is called *GBest-Selected*, and it can be expressed mathematically as follows: $t_{\text{eval}} = G_l$ with $l = \arg \max_{j=1..C} f_j(t'_{\text{eval}})$, where t'_{eval} is the number of time steps that the population has been evaluated. This approach leads to a faster convergence of the population than when using *GMaximum*; however, if the length of the best chromosome in the previous generation happen to be very large, this would lead to a larger number of time steps using only this strategy. This would be disadvantageous for the agent if the strategy actually leads to losses instead of profits for the current returns. A better approach is to choose the the number of time steps needed for evaluation according to the length of the best chromosome at every time step t . This approach is called *GBestCurrent*, and can be expressed mathematically as follows: $t_{\text{eval}} = G_l$ with $l = \arg \max_{j=1..C} f_j(t)$.

Note that the last two approaches have the disadvantage that they do not assure that all genes of all chromosomes are being evaluated; however, from our point of view, good chromosomes would lead to larger fitness than bad ones from the very beginning of the evaluation. It can happen that by coincidence the cycle of the returns match exactly a small number of good genes in bad chromosomes; however, on the long run only the good chromosomes would subsist. Unless otherwise indicated, we assume on the following that the approach *GBestCurrent* is being used for the evaluation of the population.

3.3.2 Elitist and Tournament Selection

Once the time has come to select a new population, the question is: *how to determine a new population?*

After calculating the fitness of each chromosome according to Eq. (2), we first find the best chromosomes from the old population by applying elitist and tournament selection of size two. Elitist selection considers the best s percentage of the population which is found by ranking the chromosomes according to their fitness. These best chromosomes are directly transferred to the new population. Afterwards, a tournament selection is done by randomly choosing two pairs of two chromosomes from the old population and then selecting from each pair the one with the higher fitness. These two chromosomes are not simply transferred to the new population, but undergo a transformation based on the genetic operators' crossover and mutation.

3.3.3 Crossover and Mutation Operators

Once two chromosomes have been selected by means of the tournament selection, a simple crossover operator would be one that exchanges genetic information between the two chromosomes, whatever their sizes, by finding the cross-point with respect to the size of the shortest chromosome. More in detail, this is done by selecting randomly from the shortest chromosome the cross point or cut point, c_p , and with probability $p = 0.5$ to exchange the genetic material above or beyond this cross point in the shortest chromosome with its counterpart in the largest chromosome. However, those genes in the largest chromosome beyond the length of the shortest chromosome would be disregarded.

The limitations of conventional crossover in GA with variable length has already been addressed by some authors [15], where neural networks or hierarchical tree-structures are used to determine which genes should be exchanged between the chromosomes. However, for the purpose of this paper and for the sake of simplicity, we propose a modification of the standard GA crossover operator that better suits our demands.

Thus, we propose the use of a crossover operator called *Proportional Exchange Crossover* (PEC) operator, which basically shrinks or stretches the genetic information between the pair of chromosomes proportionally to their length. Basically, the crossover operator PEC first randomly selects the range of genetic information to be exchanged between two chromosomes and contracts(extends) the genetic information from the largest(shortest) to the shortest(largest) chromosome, respectively.

More in detail, the Algorithm 1 shows the PEC algorithm for all pair of parent-chromosomes being selected via tournament selection. Note that a chromosome j is saved in an array with indexes in the range 0 to $G_j - 1$.

Algorithmus 1 : Proportional Exchange Crossover (PEC) operator

```

1 foreach pair of parent-chromosomes do
2   determine the size  $G_s$  of the shortest parent-chromosome  $pa_s$ 
3   find the cross-point,  $cp_s \in \mathbb{Z}$ , for the shortest parent-chromosome:  $cp_s \sim U(0, G_s - 1)$ 
4   determine the size  $G_l$  of the largest parent-chromosome  $pa_l$ 
5   find, the cross-point  $cp_l \in \mathbb{Z}$  for the largest parent-chromosome:  $cp_l = \frac{G_l cp_s}{G_s}$ 
6   determine the proportion  $R \in \mathbb{Z}$  between the two chromosomes' sizes:  $R = \frac{cp_l}{cp_s}$ 
7   create two arrays,  $ch_s$  and  $ch_l$ , for the short and large children-chromosomes
8   with equal probability choose the side for the crossover operation
9   if crossover on the left side then
10    extend the genetic material from  $pa_s$  and copy it to  $ch_l$  as follows:
11    for  $m = 0$  to  $cp_s - 1$  do
12      for  $n = 0$  to  $R - 1$  do
13         $ch_l[m * R + n] \leftarrow \text{extend}(pa_s, m, R)$ 
14      end
15    end
16    contract the genetic material from  $pa_l$  and copy it to  $ch_s$  as follows:
17    foreach  $m = 0$  to  $cp_s - 1$  do
18       $ch_s[m] \leftarrow \text{contract}(pa_l, m, R)$ 
19    end
20  else
21    extend as in line 1 but for the range  $m = cp_s$  to  $G_s - 1$ .
22    contract as in line 1 but for the range  $m = cp_s$  to  $G_s - 1$ .
23  end
24  copy directly the rest genetic material from the parents to the children chromosomes.
25 end

```

Note, that different functions could be considered for the transformation of the genetic material between chromosomes with different length. For simplicity, we consider in our computer experiments a PEC operator based on averaging and copying the genetic material of the parent-chromosomes. This means that in Algorithm 1, we consider for our implementation of GACE: in line 13 the function $\text{extend}(pa, m, R) = pa[m]$, which simply copies the genes from the short parent-chromosome to the large child-chromosome; and in line 18 the function $\text{contract}(pa, m, R) = 1/R \sum_{i=m}^{m+R} pa[i]$, which performs an average over the genetic material. A more interesting option for this transformations could be based on the dynamic time warping algorithm [35] which is usually used for the calculation of the similarity between two signals. With some modifications, this algorithm could be used to stretch or to shrink the genetic material proportionally to the original material; however, this is far from the scope of this paper.

To illustrate how the PEC operator works, we show in Figure 1 a pictorial representation of PEC applied to the left side of the cross-point. In this example the cross-point of the shortest chromosome is $cp_s = 3$. Consequently, using line 5 in Algorithm 1, we find that the cross-point for the largest chromosome corresponds to $cp_l = 6$. In this example the genes to the left of the shortest “parent” chromosome are generalized into the largest “child” chromosome, whereas the genes to the right of the cross-point are directly copied into the shortest “child” chromosome. The same occurs for the genes in the largest “parent-chromosome” with the main difference that the value of the genes to the left are averaged and not generalized. If the right side of the crossover is selected, we determine in the same manner the cross-points in the “parent” chromosomes and we obtain the gene values for the “child” chromosomes.

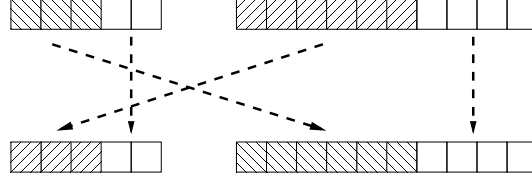


Figure 1: Example of the *Proportional-sized Exchange Crossover* (PEC) operator. With probability $p = 0.5$ the left side of the cut point is selected for exchange.

Now, to make sure that a population with chromosomes of diverse lengths is present, we introduce a mutation operator for the length of the chromosome, G_j . For this, a new length is drawn randomly and the genetic information of the chromosome is proportionally scaled to the new length. In other words, this operator mutates the length of the chromosome G_j with probability p_l leading to a new enlarged or stretched chromosome. The algorithm used for the mutation of the length of the chromosome is based on the same principle as the PEC operator.

Thus, the combination of the PEC operator and the mutation in the chromosomes’ length may help to determine the optimal investment proportions and the periodicity (or patterns) of the returns, respectively.

After the crossover and length-mutation operators are applied, the typical gene-mutation operator is applied. This means that with a given mutation probability $p_m \in U(0, 1)$, a gene is to be mutated by replacing its value by a random number from a uniform distribution $U(q_{\min}, q_{\max})$.

Summing it up, given a population with C chromosomes, to obtain a new generation of chromosomes one needs to do the following:

- I apply the elitist operator to select the best s percent of the population which are directly included in the new population.
- II the tournament selection operator is applied to the current population to select two “well-fitted” parents.
- III with probability p_c , the PEC crossover operator is applied to the two selected parent-chromosomes yielding two children-chromosomes.
- IV with probability p_l , we apply the length-mutation operator to the two children to ensure length diversity in the new population.
- V with probability p_m , the gene-mutation operator is applied to the two children which are then included in the new population.
- VI and finally, steps II to V are repeated until the new population has the same number of chromosomes as the original population.

3.3.4 Strategy Selection and Initialization

Once a new population has been obtained, we need to answer the following question: *how does the agent update its actual investment proportion, $q(t)$?*

For every new generation, the agent takes the set of strategies g_{jk} from the chromosome j with the largest fitness in the previous generation.

$$q_i(t) = g_{lk} \quad \text{with } l = \arg \max_{j=1, \dots, C} f_j; \quad k \equiv t \bmod G_l \quad (3)$$

For the initialization, each g_{jk} is assigned a random value drawn from a Uniform distribution: $g_{jk} \sim U(q_{\min}, q_{\max})$. And the length of the chromosomes can be set initially to a fixed number of genes or it can be determined randomly. For the latter, each G_j is initialized with an integer random value drawn from a Uniform distribution, where G_{\max} is the maximal allowed chromosome length.

4 Experimental Results

In this section we systematically analyze the performance of the strategies presented previously. For this, we present in Sec. 4.1 the environment for the agent, i.e. the returns, and for each environment we first investigate the parameter tuning of the Genetic Algorithm by means of increasing systematically the complexity of the operators and the environment and we finally compare the performance of the investment strategies presented in this paper.

4.1 Artificial Returns

First, we consider artificially generated returns, which are driven by the following dynamics:

$$r(t) = (1 - \sigma) \sin \left(\frac{2\pi}{T(t)} t \right) + \sigma \xi, \quad (4)$$

where the amplitude of the sinusoidal function depends on the amplitude noise level $\sigma \in (0, 1)$, and ξ corresponds to a random number drawn from a Uniform distribution, $\xi \in U(-1, 1)$. The periodicity of the returns depends on the current time step and would be present for a number of t' time steps, for initial $t' = 0$, we have:

$$\text{if } t < t' \Rightarrow T(t) = T(t-1) \quad (5)$$

$$\text{else} \Rightarrow \begin{cases} T(t) = \tilde{T} \\ t' = t + \tilde{t}, \end{cases} \quad (6)$$

where both \tilde{T} and \tilde{t} are random numbers drawn from the Uniform distributions $U(0, T_{\max})$ and $U(0, t_{\max})$, respectively.

Thus, σ accounts for the fluctuations in the market dynamics on the amplitude of the RoI; T_{\max} accounts for the largest possible periodicity and t_{\max} accounts for the maximal number of time steps a periodicity can elapse.

Figure 2 shows an example of the RoI for different noise level σ .

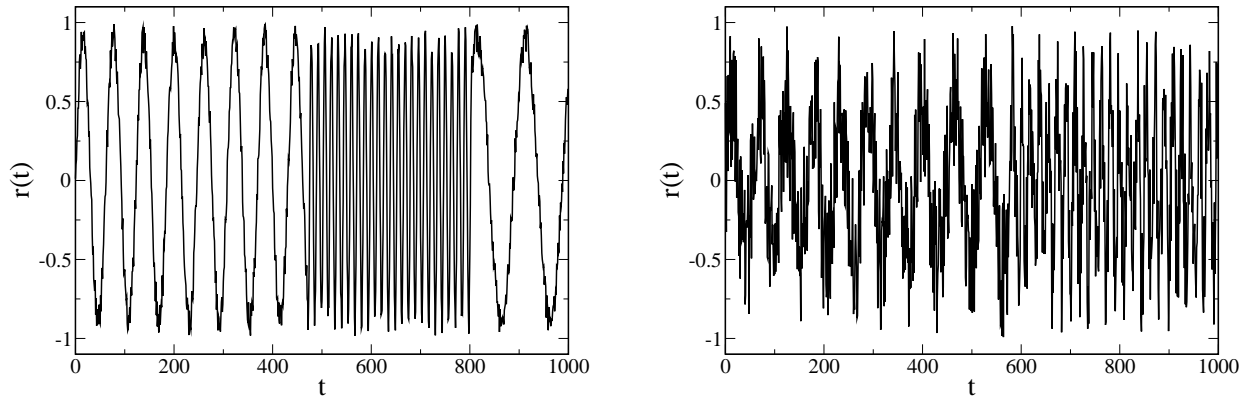


Figure 2: Periodic RoI, $r(t)$, Eq. (4) for different amplitude fluctuations: (left) $\sigma = 0.1$, and (right) $\sigma = 0.5$. Further parameters: $T_{\max} = 100$ and $t_{\max} = 1000$.

4.2 Reference Strategies

For comparison purposes, we present in this section different strategies which are used as a reference for the performance of the adaptive strategy. Note that we could have considered other type of strategies which may lead to a more complete study. However, our main goal is to show the performance of the adaptive strategy GACE comparing it with respect to the performance of other strategies for the same investment scenario. The reference strategies that we selected may be less complex than the adaptive strategy, however, they may have access to more information about the scenario.

4.2.1 Strategies with Zero/Complete knowledge

For comparison reasons, we present in this section two strategies which represent two simple behaviors for an agent; the first one, called *Constant-Investment-Proportion* (CP), assumes a simple constant minimal investment proportion, whereas the second one, called *Square-Wave* (SW), increases/decreases the investment proportion accordingly to the periodicity of the returns. In our approach, the CP strategy represents the agent with zero knowledge and zero-intelligence, whereas the strategy SW represents the agent with complete knowledge of the environment.

Constant Investment Proportion

The simplest strategy for an agent would be to take a constant investment proportion for every time step, for simplicity we call this strategy CP:

$$q(t) = q_{\min} = \text{const.} \quad (7)$$

Since the value of $q(t)$ is always fixed, this is not really a “strategy”, but a fixed attitude toward risk and it plays a role in physics inspired investment models. For this model, it has been shown that if a budget-barrier or incomes are assumed, the budget of the agent reaches a stationary distribution in the course of time and the tail of the distribution can be described with a power law function; see [23, 38].

Square Wave Strategy

The second strategy we consider as a reference is the strategy called *Square-Wave* (SW). An agent using

this strategy invests q_{\max} during the positive cycle of the periodic return, i.e. where the return has a larger probability to be positive than negative, and invests q_{\min} otherwise.

It is important to notice that this **reference strategy** assumes that the agent **knows** in advance the **periodicity**, T , of the returns.

For the sake of completeness, we describe this strategy as follows:

$$q(t) = \begin{cases} q_{\max} & t \bmod T < T/2 \\ q_{\min} & \text{otherwise.} \end{cases} \quad (8)$$

Other strategies with a similar behavior to this previous may be proposed. For example, the strategy to increase the investment proportion only for the time steps where returns are certain to be positive and not for the whole positive period of the returns. More in detail, this would mean that the agent is considering the worst scenario, which analytically can be expressed as follows:

$$r_w(t) = (1 - \sigma) \sin\left(\frac{2\pi}{T(t)} t\right) - \sigma. \quad (9)$$

It can be shown that by solving $r_w(t) = 0$ for t , the range of time steps in a cycle for which the returns are certain to be positive is determined by:

$$[\epsilon, (T/2) - \epsilon], \quad (10)$$

where:

$$\epsilon = -\frac{T}{2\pi} \arcsin\left(\frac{\sigma}{\sigma - 1}\right). \quad (11)$$

We illustrate this in Figure 3 (left). Note that large investments should be performed only in region 2 where returns are certain to be positive, whereas in regions 1 and 3 the agent may perform only moderate investments, and in the other regions the agent should in general avoid any investment.

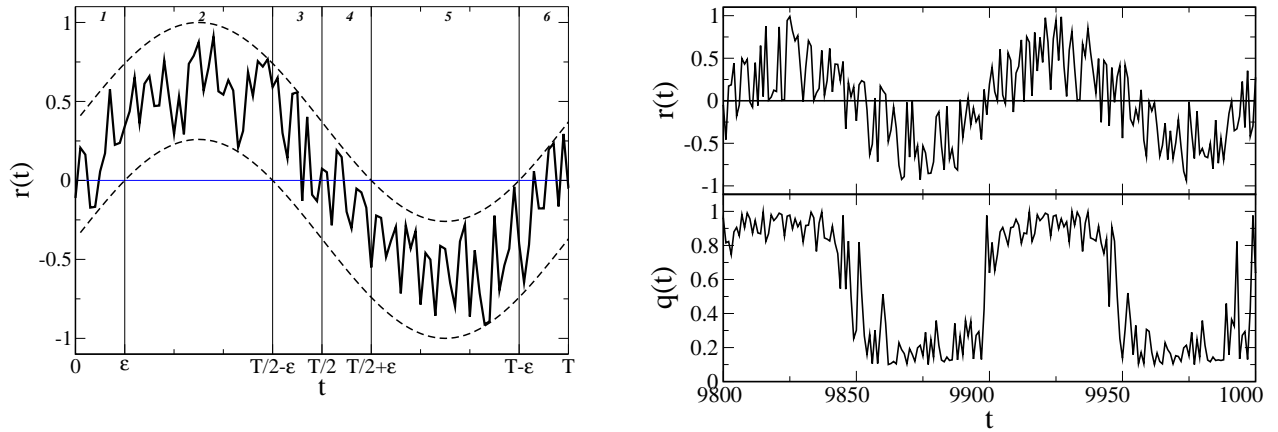


Figure 3: Return and investment strategy: (left) Different regions for the returns based on the noise level. Large investments are recommended in region 2 because returns are certain to be positive, and (right) return, $r(t)$ and the investment proportion, $q(t)$, using the strategy GACE after $t = 10^4$ time steps. Both for returns with $T = 100$ and $\sigma = 0.5$

In order to elucidate the performance of these previous possibilities for the SW strategy, let's consider returns with no noise, $r(t) = \sin(2\pi t/T)$, and the agent's wealth dynamic in Eq. (1) with initial budget $x(0) = 10$. For an agent using the investment strategy SW in Eq. (8) with $q_{\min} = 0.1$ and $q_{\max} = 1.0$, we find that the maximum possible budget for $T = 100$ after $t = 100$ time steps is $x(100) = 6.746 \times 10^9$. Now, assuming that returns have some noise, σ , for $T = 100$ we find that for $\sigma = 0.1$ and after $t = 100$ time steps, the strategy SW leads to the budget $x(100) = 1.489 \times 10^9$. Note that following Eq. (10) for $T = 100$ and $\sigma = 0.1$, returns are certain to be positive for time steps in the range $t \bmod T \in [2, 48]$. Now, if we assume an agent with the investment strategy to be a constant investment proportion of $q = 1.0$ for time steps in this range and $q = 0.1$ otherwise, it can be shown that at the end of a cycle this leads to a budget of $x(100) = 1.226 \times 10^9$, which is less than the budget obtained using SW. Furthermore, for $\sigma = 0.5$, if (despite the noise) the agent uses again the SW strategy, this leads after $t = 100$ time steps to the budget $x(100) = 1.373 \times 10^6$, whereas if the agent has an investment proportion of $q = 1.0$, for those time steps in the ranges: $t \bmod T = \{[2, 48], [10, 40], [24, 26]\}$ and $q = 0.1$ otherwise. It can be shown that these previous may lead to the budgets $x(100) = \{1.23 \times 10^6, 116015, 13.41\}$, respectively. This means that even returns have large noise, the best strategy is to increase the investment proportion once the returns are more probable to be positive than negative and not only for the returns that are certain to be positive.

4.2.2 Strategy based on Technical Analysis

We decided to include in our study a strategy based on *technical analysis* methods, which are frequently used by traders to forecast returns.

For simplicity, we chose the *Moving Least Squares* (MLS) technique and we avoided strategies based on *Moving Averages* (MA). When the latter is used, there is a 'lag' in time with respect to the current return. This causes an underestimation/overestimation for increasing/decreasing returns.

For the strategy MLS, we consider an agent with a memory size M to store previous received returns, and basically this strategy fits a function to the previous M returns, to estimate the next return, $\hat{r}(t)$. For simplicity, we chose this function to be a linear trend-line, which is found by minimizing the distance of this function to the stored returns.

Noteworthy, once the next return has been estimated, the agent still needs to perform the corresponding adjustment of the investment proportion. For this, we consider that the agent has a *risk-neutral* behavior, i.e. for small or large fluctuations of the RoI, the agent updates its investment proportion according to the expected return only. In this approach, the value of $q(t)$ is updated as follows:

$$q(t) = \begin{cases} q_{\min} & \hat{r}(t) \leq q_{\min} \\ \hat{r}(t) & q_{\min} < \hat{r}(t) < q_{\max} \\ q_{\max} & \hat{r}(t) \geq q_{\max} \end{cases} \quad (12)$$

where $q_{\min}, q_{\max} \in [0, 1]$. In other words, the agent invest q_{\min} if the estimated return for the next time step is negative or zero, otherwise it invests proportional to the estimated next return.

4.3 Results for RoI with fixed periodicity

To elucidate the performance of the adaptive strategy proposed in this paper and the reference strategies previously presented, we start with a simple scenario where returns have a fixed periodicity. In Section 4.4 we consider a more challenging scenario where returns have a changing periodicity.

First, we assume that the parameters of a strategy lead to an optimal performance, if it leads to the *maximum total budget* that can be reached with this strategy during a complete period of the returns. When evaluating the

strategies, we have to consider that their performance is also influenced by stochastic effects and. In the case of the strategy GACE we also have to account for the different possible strategies that may evolve. This means that we have to average the simulation over a large number of trials, N , where each trial simulates an agent acting independently with the same set of strategy parameters. More in detail, the performance of an agent in a single trial corresponds to the average budget at the end of each RoI's period, T ; thereafter, an average over a number of trials is performed to diminish noise effects. For convenience, the total budget has been normalized by the number of cycles or periods of the RoI, I . This is done, because if the strategy performs well, the budget of the agent may reach very high values. This occurs because in the dynamics of eqn. (1) the budget could possibly be doubled at each time step, if an appropriate $q(t)$ and $r(t)$ are provided. In the computer simulations this would lead to numerical overflows, therefore we have chosen to reinitialize the budget after each cycle of the RoI, which applies to all simulations, to ensure comparison.

4.3.1 GACE Parameter Tuning

In the following, we want to find the parameter values that lead to larger fitness and budget values for the strategy GACE. It is well known that the configuration of most metaheuristic algorithms requires both complex experimental designs and high computational efforts. For finding the best parameters for the GA, a software called +CARPS (*Multiagent System for Configuring Algorithms in Real Problem Solving*) [30] was used. It consists of autonomous, distributed, cooperative agents that search for solutions to a configuration problem, thereby fine-tuning the metaheuristic's parameters.

The GA was configured for periodic returns with $T = 100$ and different level of noise: $\sigma = 0.1$, and $\sigma = 0.5$. In this process, four GA parameters were optimized: the population size C , the crossover probability p_c , the mutation probability p_m , and the elitism size s . Their intervals of definition, in which the most acceptable GA configurations should be found, were set as follows: $C \in \{50, 100, 200, 500, 1000\}$, $p_c \in [0.0, 1.0]$, $p_m \in [0.0, 1.0]$, and $s \in [0.0, 0.5]$.

When configuring, agents in +CARPS apply a Random Restart Hill-Climbing approach and they exchange best-so-far solutions during this process. Furthermore, the evaluation of the GA with a particular configuration is repeated five times in order to cope with its stochastic nature. According to the fitness, we show in Table 1 the best obtained configuration for the GA in the periodic returns previously mentioned.

Table 1: GACE's best parameter values for RoI with fixed T .

C	p_c	p_m	s
1000	0.7	0.01	0.3

For the sake of completeness, we show in Figure 4 (left) the evolution in the course of generations of the average fitness of the chromosomes in the population for different mutation rates.

Moreover, in Figure 4 (right) shows for $C = 1000$ chromosomes, the evolution of the average fitness and the largest fitness in the course of generations. Observe that the rate $p_m = 0.01$ used for Figure 4 (right), leads to larger average fitness in the population than for less or more mutation rate showed in Figure 4 (left); however, note that the fitness of the best chromosome when using $p_m = 0.001$ is almost as well as for $p_m = 0.01$.

Note that in Figure 4 (right), for the first 100 generations the best chromosome performs much better than all the chromosomes in average; however, after 100 generations we can see that the performance of the population converges to the performance of the best chromosome. Now, consider again Eq. (2) and replace: g_{jk} with $q(t)$, and G_j with T . If we consider returns for $t = 100$ time steps with periodicity $T = 100$ and no noise, it can be shown that the strategy SW would lead to a fitness of $f(\tau) = 28.63$. Note that this is not much larger than the fitness obtained with GACE.

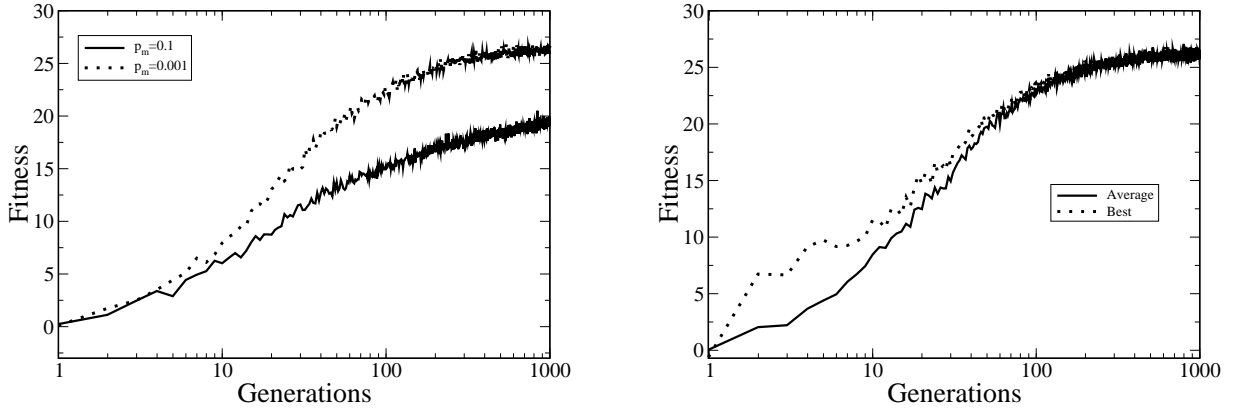


Figure 4: Performance graphs for the GACE strategy in the course of generations for $C = 1000$ chromosomes: (left) average fitness for different mutation probabilities, p_m ; and (right) convergence of the population showing the average fitness against the fitness of the best chromosome for $p_m = 0.01$. Further parameters $T = 100$ and $\sigma = 0.1$.

Now, to better illustrate the set of investment strategies that are being obtained using GACE, we show in Figure 3 (right) the RoI and the investment proportions obtained after a number of time steps for returns with relative large noise. For the reader with background in signal processing techniques, Figure 3 may sound familiar as it resembles to those figures obtained when using matched filters for signal recovery, see [40] for more on matched filters.

4.3.2 Performance Comparison

In order to assure fair comparison between the strategies, we need to find the most proper parameter values for the strategies. Note that for both strategies CP, Eq. (7) and SW, Eq. (8), we don't need to tune any parameters. However, for the strategy MLS, Eq. (12), we assume that the agent has access to some information about the returns, i.e. the agent knows the periodicity, T , of the returns. This means that the agent needs to determine the most proper memory size, M , based on the known periodicity of the returns. For this, we performed some experiments for returns with different fixed periodicities, T , and no noise, $\sigma = 0$. Figure 5 (left), shows the results of these experiments, where the budget of an agent is shown for different memory sizes and for returns with different T , and no noise. According to visual impression, the most proper memory size, M , and the periodicity, T , are proportionally related by $M/T \approx 0.37$.

Moreover, if we assume returns with no noise, we can find analytically the memory size M^* that maximizes the profits. For this, we note that for a periodic return as in Eq. (4) with $\sigma = 0$, the strategy MLS estimates the next return $\hat{r}(t+1)$ as follows:

$$\begin{aligned} \hat{r}(t+1) &= \frac{\sin(\omega t) - \sin(\omega(t-M))}{\omega t - \omega(t-M)} (\omega(t+1) - \omega(t-M)) + \sin(\omega t - \omega M) \\ &= \frac{M+1}{M} [\sin(\omega t) - \sin(\omega t - \omega M)], \end{aligned} \quad (13)$$

where $\omega = 2\pi/T$. Now, by calculating the average profits $\langle r q \rangle$ for the positive cycle of the returns, we find:

$$\begin{aligned}\langle r q \rangle &= \int_0^{T/2} r(t) q(t) dt \\ &= \frac{T(M+1 - \cos(\omega M))}{4M}.\end{aligned}\quad (14)$$

Figure 5 (right) shows the resulting budgets for different memory size values when using Eq. (14). Note that we can find the memory size that leads to maximum profits by finding the derivative of $\langle r q \rangle$ w.r.t M , which is:

$$\partial_M \langle r(t) q(t) \rangle = \frac{-T \sin\left(\frac{\omega}{2} M\right)^2 + \pi M \sin(\omega M)}{2M^2}.\quad (15)$$

Thus, the memory size M^* that maximizes the profits can be calculated by solving $\partial_M \langle r(t) q(t) \rangle = 0$. Using Taylor series to the sixth order for the sinusoidal functions we end with the following expression:

$$M^* = \frac{\sqrt{\frac{3}{2}}}{\pi} T.\quad (16)$$

Consequently, for $T = 100$ the theoretical optimal memory size is $M^* = 38$, which agrees with the empirical optimal memory size shown in Figure 5 (right) for different noise levels. Relatedly, the proportion $M/T \approx 0.37$ found by means of computer simulations in Figure 5 (left), approximates pretty well the proportion found analytically $M/T = \sqrt{\frac{3}{2}}/\pi = 0.389$.

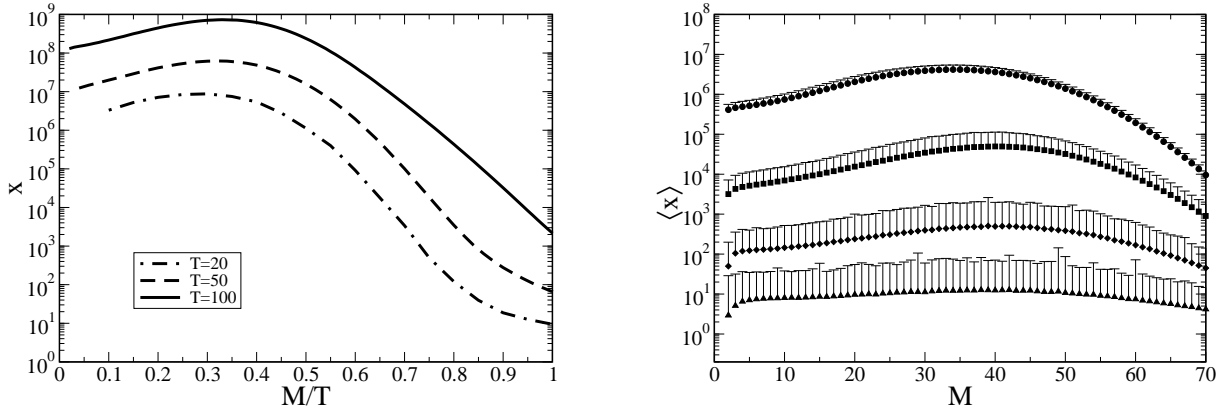


Figure 5: For strategy *MLS*, both plots show the budget obtained for different memory size M for: (left) different periodicity T and no noise, (right) different noise level σ and periodicity $T = 100$.

Comparison for fixed chromosome length

Now, we compare the performance of the adaptive investment strategy GACE, presented in Section 3, with respect to the reference strategies presented in Section 4.2. For the sake of clarity, we assume for the moment that the strategy GACE uses fixed chromosome length, i.e. $G_j = G_{\max}$. For all strategies we consider

$q_{\min} = 0.1$ and $q_{\max} = 1.0$ in our experiments. These parameter values describe the behaviour of the strategies CP, Eq. (7), and SW, Eq. (8). For the strategy MLS, Eq. (12), we use Eq. (16) to determine the optimal memory size and for the strategy GACE we use the parameters in Table 1.

As it was done previously, we generate a synthetic data set for the returns. In our experiments we assume that the agent invests in returns with periodicity $T = 100$ for different noise levels. We consider here that the length of the chromosomes is fixed to $G_j = 100$ and a new generation of chromosomes is being obtained after a number of time steps $t_{\text{eval}} = 100$. In other words, we consider for these experiments the approach GMaximum to determine the number of time steps needed to evaluate the population, see Section 3.3.1. Note that we investigate the simplest case where the periodicity of the returns perfectly maps to the length of the chromosomes and time steps for evaluation of the population. For the computer experiments, we let the agent to use one of the strategies to invest during a number of $t = 10^5$ time steps. In order to account for the randomness of the scenario, we perform the experiment for a number of $N = 100$ trials, gathering the average budget obtained for each strategy at every 100 time steps.

Figure 6 shows in a log-log plot the average budget, $\langle x \rangle$, in the course of GACE's generations, τ , for all strategies and for returns with different amplitude noise levels. As you can see, except for the GACE strategy, all other strategies have a constant budget in average over each generation. This occurs because the average of the budget was taken at every $t_s = 100$ time step which corresponds to the periodicity of the returns $T = 100$ and to the time steps to evaluate the population of chromosomes $t_{\text{eval}} = 100$, as it was specified in our experiment parameters.

More in detail, Figure 6 (left) shows that after 4, 70, and 300 generations, GACE over-performs the strategies $q = 0.1$, and MLS, respectively. And we note that GACE performs almost as well as the strategy SW after 400 generations. Moreover, the budget of the agent using GACE increases approximately according to a power law for the first 100 generations, afterwards increases logarithmically.

Figure 6 (right) shows that for large amplitude noise it takes fewer generations for GACE to over-perform the strategy $q = 0.1$, but in general more generations are needed for GACE to over-perform the other strategies. We find that the budget also increases according to a power law for the first 100 generations and afterwards increases logarithmically. We note that it would be useful to provide with a formulation to characterize the average budget in the course of generations that is obtained using the strategy GACE, however, this is left for further work.

Comparison for variable chromosome length

In this section, we investigate the case when the chromosomes do not have initially the same length as the periodicity of the returns. For this, we let the initial length of the chromosomes to be chosen at random from a Uniform distribution. This means that we need to define the range of possible length values. For our implementation of the GA to work properly, the unknown periodicity needs to be in the range of possible length values. For simplicity, in our experiments we let the range to be larger than the periodicity of the returns. However, we note that this parameter could be determined by the GA itself if we include extra genes in the chromosome to track for a proper range. Another possibility could be to determine this parameter by means of statistical properties of the returns, like the autocorrelation function or spectral density; however, both approaches are beyond the scope of this paper and are left for further work.

Now, for the case that initially the length of the chromosomes is different, if population is evaluated after a fixed number of time steps, the following question may arise: *Do the chromosomes' lengths correctly evolve to map the periodicity of the returns?*

To answer this question, we performed some computer experiments for an agent using the strategy GACE for returns with periodicity $T = 100$ and different noise levels. For these experiments, we assumed for GACE

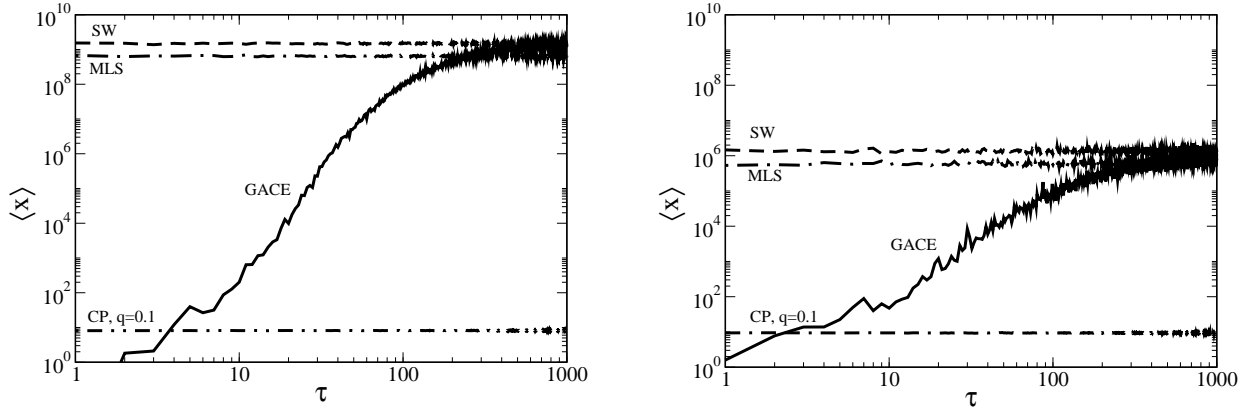


Figure 6: Average budget, $\langle x \rangle$, obtained using different investment strategies in the course of generations τ of the strategy GACE, for returns with periodicity $T = 100$ and amplitude noise: (left) $\sigma = 0.1$ and (right) $\sigma = 0.5$.

the parameter values specified in Table 1, and of special interest, we now consider that the initial chromosomes' length is drawn randomly from a Uniform distribution with range of values $(1, G_{\max})$, with $G_{\max} = 500$. Furthermore, we assume that the evaluation of the population, leading to a new generation of chromosomes, is performed every $t_{\text{eval}} = 500$ time steps, i.e. we consider for these experiments the approach G_{\max} to determine the number of time steps needed to evaluate the population, see Section 3.3.1.

Figure 7 shows the probability distribution of the length of the best fitted chromosomes for different noise levels and for different generations, $\tau = \{5, 100\}$.

It is clear that after five generations most of the chromosomes' length have properly matched the periodicity of the returns. Interestingly, chromosomes with lengths proportional to a multiple of the periodicity are also frequent; however, the probability decreases for larger multiples of the real periodicity, which is a consequence of the better adaptation of smaller chromosomes which have found more quickly the most proper investment proportions.

4.4 RoI with changing periodicity

In the previous section, we deal with a stationary environment, now in this section we tackle a non-stationary environment. For comparison reasons, we start presenting some computer experiments for the strategy GACE using the parameters for a stationary environment, shown in Table 1, now for returns with non-fixed periodicity.

Figure 8 (top) shows the evolution of the average budget in the course of time for an agent with the strategy GACE investing in returns with changing periodicity and different noise levels. For the sake of clarity, we include in Figure 8 (bottom) the corresponding periodicities of the returns for each time step.

Note that for these experiments we use the selection approach GBestCurrent, see Section 3.3.1. Also note that in order to avoid overflows, see Section 4.3, the budget of the agent is reinitialized to the initial budget every time the periodicity of the returns changes. From the dynamics of the returns, Eq. (4), it can be seen that a change of periodicity is not performed exactly at the end of a period but at any time step. This is the reason for large increases or decreases of budget each time the periodicity of the returns changes.

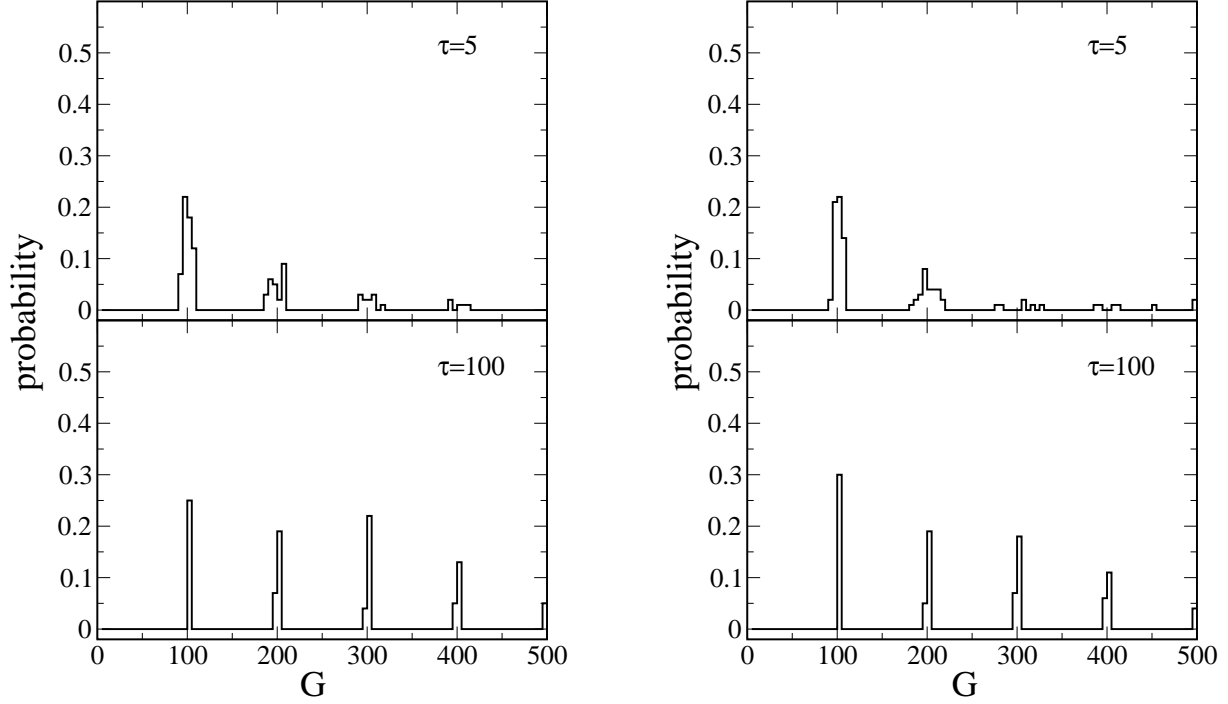


Figure 7: Probability distribution of the length of the best fitted chromosomes for generations $\tau = \{5, 100\}$ for $N = 50$ trials. Returns with periodicity $T = 100$ and amplitude noise: (left) $\sigma = 0.1$ and (right) $\sigma = 0.5$.

4.4.1 GA Parameter Tuning

As we did before, we address the problem of determining the most proper parameter values for GACE, now for returns with changing periodicity. For this, we performed some experiments and determined empirically the most proper parameter values for GACE when using the approach GBestCurrent, these results are shown in Table 2.

Table 2: GACE's best parameter values for RoI with changing T .

C	p_c	p_m	s	p_l
1000	0.5	0.001	0.3	0.5

Note that with respect to Table 1, if the parameter values in Table 2 are used, the crossover and mutation operators are less probable to occur when recombining two parents. However, this is covered by a surprising large probability of mutation on the length of a chromosome.

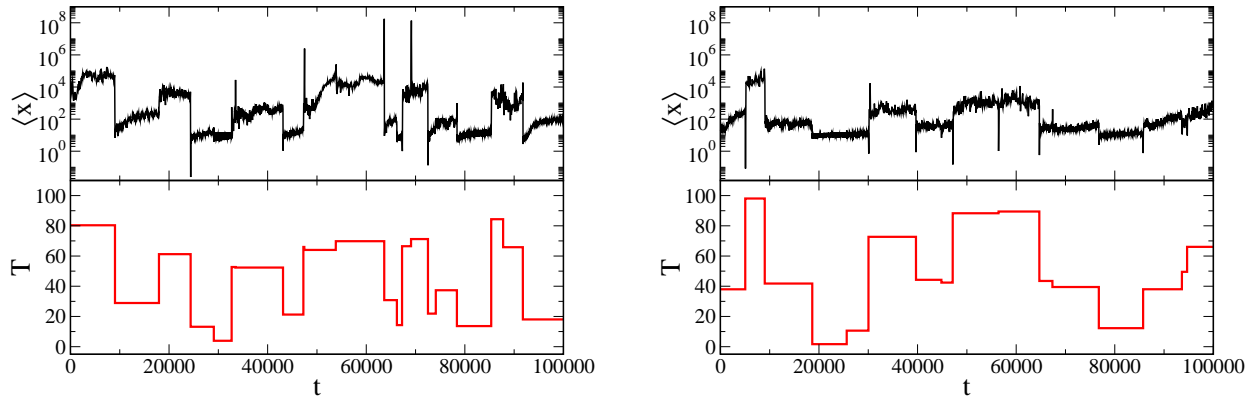


Figure 8: (top) Average budget in the course of time for $N = 50$ trials for an agent using the strategy GACE with parameter values as in Table 1 and the parameter values $G_{\max} = 200$ and $p_l = 0.1$. (bottom) Periodicity of the returns in the course of time, Eq. (4), with parameters: $T_{\max} = 100$ and $t_{\max} = 10^4$. Both for different amplitude noise: (left) $\sigma = 0.1$ and (right) $\sigma = 0.5$.

4.4.2 Performance Comparison

In this section we investigate the performance of the adaptive strategy with respect to the reference strategies in a non-stationary scenario. For this, we performed some computer experiments for returns with changing periodicity and different noise level. As we did in the previous sections we assumed for all strategies the parameter values $q_{\min} = 0.1$ and $q_{\max} = 1.0$. Moreover, for the strategy MLS we used Eq. (16) to calculate the memory size, M . And for the strategy GACE we used the parameter values listed in Table 2 and the length of a chromosome in the range $G_j \in (1, G_{\max})$, with $G_{\max} = 200$.

We show in Figure 9 (top) the evolution of budget, and (bottom) the corresponding periodicity of the returns, Eq.(4), both in the course of time for the different investment strategies and different noise levels. It is clear that the best strategy is for both cases the strategy SW, following the strategy MLS; however, note that both strategies have total and partial knowledge about the dynamics of the returns, respectively. As we mentioned previously, the strategy SW, Eq. (8), knows the dynamics of the stylized returns and increases the investment proportion for the positive periods and decreases it for the negative. On the other hand, the strategy MLS, Eq. (12)), knows the periodicity T of the returns, which is used to calculate the most proper memory size by means of Eq. (16). This previous knowledge gives some advantage to these strategies over the strategy GACE, which only needs the specification of G_{\max} . We note that the strategy GACE evolves quite fast, yielding a set of investment strategies with a clear tendency to lead more gains than losses. This particularly is shown for long-lasting periodicities, where an ever increasing growth of budget is observed. Interestingly, the strategy GACE performs much better than the reference strategy CP and performs on certain occasions as good as the strategy MLS, particularly for returns with small noise.

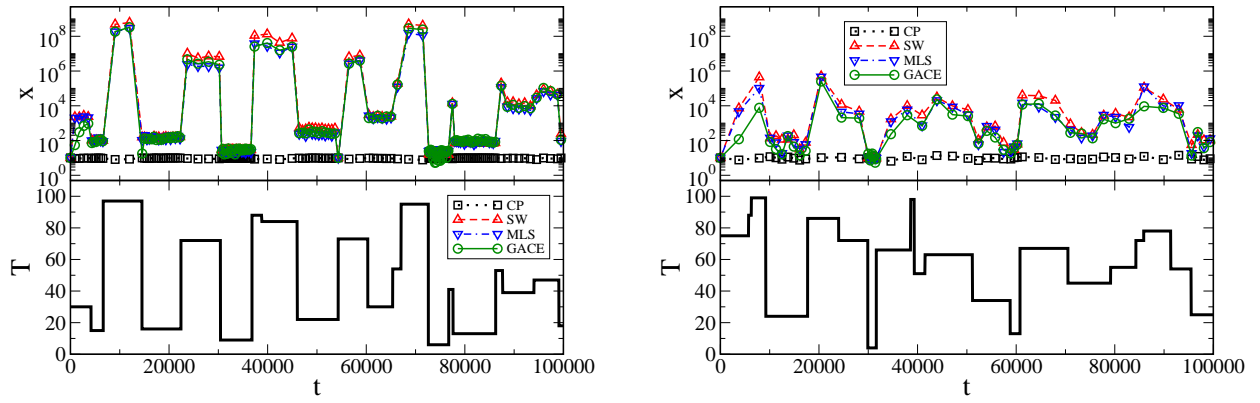


Figure 9: (top) Budget in the course of time for different strategies, assuming $q_{\min} = 0.1$ and $q_{\max} = 1.0$ for all strategies. For MLS we used Eq. (16) to calculate the memory size, M , and for GACE the parameters shown in Table 2. (bottom) Periodicity of the returns in the course of time, Eq. (4), with parameters: $T_{\max} = 100$, $t_{\max} = 10^4$, and amplitude noise: (left) $\sigma = 0.1$ and (right) $\sigma = 0.5$.

5 Conclusions

In this paper, we presented a simple investment model and some investment strategies to control the proportion of investment in periodic environments. The novelty of this paper is in the adaptive investment strategy here proposed, called *Genetic Algorithm for Changing Environments* (GACE), which is a new approach based on evolution for the correct mapping of investment proportions to patterns that may be present in the returns. We analyzed the performance of GACE for different scenarios, and compared its performance in the course of time with respect to other strategies that were used as a reference. We showed that after a given number of time steps, the strategy GACE reaches a set of investment strategies that can over-perform simple strategies like those that invest always a constant investment proportion. We showed that even though the strategy GACE has no knowledge of the dynamics of the returns, it may lead to large gains, performing as well as other strategies with some knowledge. This particularly is shown for long-lasting periodicities, where an ever increasing growth of budget was observed. This means that in the presence of long-lasting periodicities, the longer the agent uses the adaptive strategy the largest the profits per cycle.

In this study, we used artificial generated stylized returns, which are based on a sinusoidal function; however, it can be shown that for other type of periodic functions, the GA would eventually find the most proper strategy in the same way that for the sinusoidal function. Despite the fact that the strategy GACE proposed in this paper was mainly used to find the most proper set of investment proportions for an investment scenario, it is important to note that this strategy can be applied to other kind of scenarios. For example, scenarios where the agent has to control other kind of resources, like energy, time consumption, etc.

Further work includes the analysis of the performance of the strategy GACE for real returns, and to compare the performance of GACE with other similar approaches like Genetic Programming techniques, Neural Networks, and Reinforcement Learning. Useful, would be to extend also this approach for optimal portfolio diversification, where a large number of algorithms have been proposed, which deal with the research areas of optimization, stochastic simulation and decision theory.

Finally, we note that the proposed adaptive investment strategy may be interesting for the research area of pattern recognition of time series. By making proper changes in the fitness function, a useful algorithm could be obtained for the detection and measurement of periodic signal in time series.

Acknowledgements

We thank Prof. Frank Schweitzer for his advice during these investigations. We also thank Prof. Hans-Dieter Burkhard for his useful comments and suggestions and Dr. Dagmar Monett for providing us the program +CARPS (*Multiagent System for Configuring Algorithms in Real Problem Solving*).

References

- [1] Alvarez, A.; Orfila, A.; Tintore, J. (2001). DARWIN: An evolutionary program for nonlinear modeling of chaotic time series. *Computer Physics Communications* **136**(3), 334–349.
- [2] Arrow, J. K. (1965). *Aspects of the Theory of Risk Bearing*. Helsinki: Helsinki.
- [3] Artzner, P.; Delbaen, F.; Eber, J.-M.; Heath, D. (1999). Coherent Measures of Risk. *Math. Finance* **9**(3), 203–228.
- [4] Bak, P.; Norrelykke, S. F.; Shubik, M. (1999). The Dynamics of Money. *Physical Review E* **60**(3), 2528–2532.
- [5] Branke, J. (1999). Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems. In: *Proceedings of the Congress on Evolutionary Computation CEC'99*. vol. 3, pp. –1882.
- [6] Dawid, H. (1999). *Adaptive learning by genetic algorithms: Analytical results and applications to economic models*. Berlin: Springer, revised second edn.
- [7] Drake, A. E.; Marks, R. E. (2002). Genetic Algorithms in Economics and Finance: Forecasting Stock Market Prices and Foreign Exchange - A review. In: S.-H. Chen (ed.), *Genetic Algorithms and Genetic Programming in Computational Finance*, Dordrecht: Kluwer Academic. pp. 29–54.
- [8] Farmer, J. D. (2001). Toward Agent-Based models for Investment. In: *Benchmarks and Attribution Analysis*. Association for Investment and Management Research, pp. 61–70.
- [9] Farmer, J. D.; Patelli, P.; Zovko, I. I. (2005). The Predictive Power of Zero Intelligence in Financial Markets. In: *Proceedings of the National Academy of Sciences*. vol. 102, pp. 2254–2259.
- [10] Forrest, S. (1996). Genetic algorithms. *ACM Computing Surveys* **28**(1), 77–80. ISSN 0360-0300.
- [11] Geibel, P.; Wyszotzki, F. (2005). Risk-Sensitive Reinforcement Learning Applied to Control under Constraints. *Journal of Artificial Intelligence Research* **24**, 81–108.
- [12] Gode, D. K.; Sunder, S. (1993). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy* **101**, 119–137.
- [13] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Mass., USA: Addison-Wesley.

- [14] Grefenstette, J. J. (1992). Genetic algorithms for changing environments. In: R. Manner; B. Manderick (eds.), *Parallel Problem Solving from Nature 2*, Amsterdam: North Holland: Elsevier. pp. 137–144.
- [15] Harvey, I. (1992). The SAGE Cross: The mechanics of Recombination for Species with Variable-length Genotypes. *Parallel Problem Solving from Nature 2*, 269–278. North-Holland.
- [16] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: The University of Michigan Press.
- [17] Jiang, R.; Szeto, K. Y. (2003). Extraction of Investment Strategies based on Moving Averages: A Genetic Algorithm Approach. In: *CIFER'03*. Hong Kong: IEEE Press, pp. 403–410.
- [18] Kahneman, D.; Tversky, A. (1979). Prospect Theory of Decisions under Risk. *Econometrica* **47**, 263–291.
- [19] Kahnemann, D.; Riepe, M. W. (1998). Beliefs, preferences and biases investment advisors should know about. *Journal of Portfolio Management* **24**(4).
- [20] Kelly, J. L. (1956). A new Interpretation of Information Rate. *The Bell System Technical Journal* .
- [21] Kesten, H. (1973). Random difference equations and renewal theory for products of random matrices. *Acta Math.* **131**, 207–248.
- [22] LeBaron, B. (2000). Agent-based computational finance: Suggested readings and early research. *Journal of Economic Dynamics and Control* **24**, 679–702.
- [23] Levy, M.; Solomon, S. (1996). Power laws are logarithmic Boltzmann laws. *International Journal of Modern Physics C* **7**, 595–601.
- [24] Lux, T.; Marchesi, M. (2002). Special issue on heterogeneous interacting agents in financial markets. *Journal of Economic Behavior and Organization* **49**(2), 143–147.
- [25] Magdon-Ismail, M.; Nicholson, A.; Abu-Mostafa, Y. (2001). Learning in the presence of noise. In: S. Haykin; B. kosko (eds.), *Intelligent Signal Processing*, IEEE Press, chap. 3. pp. 120–127.
- [26] Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance* **7**, 77–91.
- [27] Marsili, M.; Maslov, S.; Zhang, Y.-C. (1998). Dynamical optimization theory of a diversified portfolio. *Physica A* **253**, 403–418.
- [28] Maslov, S.; Zhang, Y.-C. (1998). Optimal Investment Strategy for Risk Assets. *Mathematical Models and Methods in Applied Sciences* .
- [29] Michalewicz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin Heidelberg: Springer, Third, Revised and Extended edn.
- [30] Monett, D. (2004). +CARPS: Configuration of Metaheuristics Based on Cooperative Agents. In: C. Blum; A. Roli; M. Sampels (eds.), *Proceedings of the 1st International Workshop on Hybrid Metaheuristics, HM'2004, at the 16th European Conference on Artificial Intelligence, ECAI'2004*. Valencia, Spain, pp. 115–125.
- [31] Navarro, J. E.; Schweitzer, F. (2003). The Investors Game: A Model for Coalition Formation. In: L. Czaja (ed.), *Proceedings of the Workshop on Concurrency, Specification & Programming, CS & P'2003*. Czarna, Poland: Warsaw University, vol. 2, pp. 369–381.

- [32] Navarro-Barrientos, J. E.; Cantero-Alvaréz, R.; Rodríguez, J. F. M.; Schweitzer, F. (2007). Investments in random environments. *Physica A* (2007), doi:10.1016/j.physa.2007.11.029.
- [33] Neely, C.; Weller, P.; Dittmar, R. (1997). Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach. *The Journal of Financial and Quantitative Analysis* **32**(4), 405–426.
- [34] Redner, S. (1990). Random multiplicative processes: An elementary tutorial. *American Journal of Physics* **58**, 267–273.
- [35] Sankoff, D.; Kruskal, J. B. (1983). *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley.
- [36] Schulenburg, S.; Ross, P. (1999). An evolutionary approach to modelling the behaviours of financial traders. In: S. Brave; A. S. Wu (eds.), *Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference*. Orlando, Florida, USA, pp. 245–253. URL citeseer.ist.psu.edu/schulenburg99evolutionary.html.
- [37] Schulenburg, S.; Ross, P. (2001). Strength and Money: An LCS Approach to Increasing Returns. *Lecture Notes in Computer Science* **1996**, 114–137. URL citeseer.ist.psu.edu/schulenburg01strength.html.
- [38] Sornette, D.; Cont, R. (1997). Convergent Multiplicative Processes Repelled from Zero: Power Laws and Truncated Power Laws. *Journal of Physics* **1**(7), 431.
- [39] Szpiro, G. G. (1997). Forecasting chaotic time series with genetic algorithms. *Phys. Rev. E* **55**(3), 2557–2568.
- [40] Turing, G. (1960). An introduction to matched filters. *IEEE Transactions on Information Theory* .